Partitioned Multistack Evironments for Exascale Systems

John Lange
University of Pittsburgh
jacklange@cs.pitt.edu

1 Introduction

While there remains a certain amount of uncertainty about the architecture of future exascale systems, it is generally accepted that they will mark a significant departure from current hardware designs. In particular it is expected that exascale architectures will exhibit a much greater degree of heterogeneity both at the hardware level as well as in the software that is executed on each compute node. Additionally, it is likely that more functionality will need to be localized on each compute node due to power and performance constraints. As a result, exascale system software will be tasked with handling a much larger number of responsibilities than currently exist. For example, it is reasonable to anticipate that an exascale compute node will need to execute a given application's computational kernel across a number of different hardware processing components (such as GPUs, and lightweight CPU cores), while at the same time managing application data and checkpoints on local storage devices and performing in-situ analysis and visualization. Each of these tasks represent significantly different workloads and behaviors and achieving scalable performance requires an environment optimized for each of them.

Petascale system architectures have provided optimized environments for each task by providing specialized components (such as compute nodes, I/O nodes, and visualization clusters) dedicated to a specific set of functions and workloads. These components are separated not just physically but also at the system software layer, with each task running a specialized OS environment. However, with the consolidation of these tasks in exascale systems, it will no longer be possible to easily optimize each component individually and in isolation. This is of particular concern given the different, and often conflicting, requirements of the set of applications making up an exascale workflow. For instance, while lightweight kernels are suitable for computational tasks, they are not well suited for I/O or management roles. Exascale system software environments will thus need to provide a unified environment that meets the requirements for a widely divergent set of tasks.

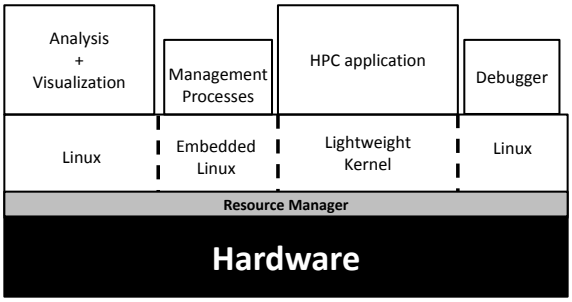

Figure 1: A hypothetical multistack environment.

The position of this paper is that no single system software stack is capable of addressing all of the requirements for every workload executing on a local node. Therefore *we believe that future exascale systems should embrace a partitioned multistack approach to system software*, wherein multiple specialized system software stacks execute in parallel and directly manage disjoint sets of hardware resources. Such an approach would enable each workload running on a single exascale node to utilize an appropriate system environment specifically tailored to its behavior. For instance, a hypothetical node would be able to host a stripped down Linux environment on a small set of cores to handle managerial functions, while a lightweight kernel hosts an HPC application on a separate set of CPU cores and in-situ analysis is performed in a heavy weight Linux environment in another resource partition. Each of these environments would manage resources directly without interference from the other system software stacks and interact only for high level coordination. An example of such a system configuration is shown in Figure 1.

2 Challenges

It is important to note that the high level approach we are proposing is not a complete departure from current ar-

chitectures of petascale systems, and many of the same mechanisms would be directly applicable to our proposed approach. Techniques such as I/O and system call forwarding are already well established to deal with the distributed nature of current supercomputing architectures, and we believe that the same approaches will still apply (albeit locally instead of over a network). However, while at a high level it might appear that our approach simply consolidates existing environments on the same node, we believe that there exist several challenges that need to be addressed and design decisions that should be revisited. In particular, resource management and allocation as well as cross stack communication pose a new set of problems not present in current system architectures.

Resource Management and Allocation Historically, supercomputing environments have avoided the overheads of multiplexing local resources, and instead focused on exposing as much of the hardware resources as possible directly to an application. Exascale systems, however, will likely remove this luxury and force system software to be much more proactive in how local hardware is managed. Our proposed approach addresses this challenge by isolating applications into separate partitions with their own OS environment to directly manage a set of resources allocated at a very coarse grained level. In our vision, a resource management layer will operate on large blocks of resources that are allocated to a given OS instance to manage on its own. We envision these resource blocks to include large chunks of contiguous physical memory, entire CPU cores or even sockets, and also entire I/O devices. Such an approach would allow the system to partition an exascale system with very little overhead, and tailor its configuration to specific applications.

Cross Stack Communication The second key challenge to address is how to enable the communication and sharing of data between OS partitions (the dashed lines in Figure 1). While existing systems have relied predominantly on network communication to achieve this, a new localized architecture will require significant changes to these approaches. New communication channels and primitives will likely be needed to fully exploit the underlying hardware. We believe that new approaches should be explored to enable communication, coordination, and data sharing between local OS partitions, in a way that exploits the connectivity of local hardware and is easily accessible to application developers.

3 Existing Work

Previous work by ourselves [2] and others [1] have explored and demonstrated the ability to provide isolated resource partitions to separate OS environments. In fact, when coupled with specialized OS architectures, such as lightweight kernels, these environments are capable of outperforming the host OS itself. For these experimental environments, the key enabler has been the availability of virtualization features in the hardware. Virtualization provides a natural approach to resource partitioning and allows lightweight resource managers to easily partition and assign large slices of the underlying hardware. Our previous work with the Palacios Virtual Machine Monitor [3] has also demonstrated the capability of deploying large scale virtualized environments on modern supercomputing platforms with minimal overhead [4, 5].

While we believe that virtualization holds great promise as a technique for managing the complexity of exascale systems, it is likely that full support for virtualization will not be present throughout all the hardware components of an exascale node. As exascale systems include a greater variety of I/O devices and specialized processing components, a complete system virtualization layer, that encompasses all of the hardware components, will most likely be difficult to achieve. For these situations it will be necessary to develop new techniques for partitioning the hardware in question and exposing it to the appropriate system stack. For some hardware devices this should be fairly straightforward as virtualization techniques exist to assign non-virtualized hardware to a guest OS, however for others (such as specialized CPU cores) this is likely to be more difficult. Nevertheless, we do not believe these issues to pose a serious hindrance to the development of our approach.

4 Beyond Exascale

As datacenter and cloud service environments continue to expand and grow, many of the solutions generated by the HPC community are finding their way into commodity systems. Many scientists and HPC application developers are also seriously considering cloud based environments as a potential substrate for their computational needs. As this occurs, many of the same problems and issues we are addressing for exascale will be faced by cloud service providers. The ability to fully partition a large scale system into fully isolated components, as described in this paper, is likely to become very useful as HPC applications begin to share systems with commodity cloud workloads.

References

[1] BUTRICO, M., DA SILVA, D., KRIEGER, O., OSTROWSKI, M., ROSENBURG, B., TSAFRIR, D., VAN HENSBERGEN, E., WISNIEWSKI, R. W., AND XENIDIS, J. Specialized execution environments. *SIGOPS Operating Systems Review 42*, 1 (Jan. 2008), 106–107.

[2] KOCOLOSKI, B., AND LANGE, J. Better than native: Using virtualization to improve compute node performance. In *Proceeedings of the 2nd International Workshop on Runtime and Operating Systems for Supercomputers* (2012), ACM.

[3] LANGE, J., DINDA, P., HALE, K., AND XIA, L. An introduction to the palacios virtual machine monitor—release 1.3. Tech. Rep. NWU-EECS-11-10, Department of Electrical Engineering and Computer Science, Northwestern University, October 2011.

[4] LANGE, J., PEDRETTI, K., DINDA, P., BRIDGES, P., BAE, C., SOLTERO, P., AND MERRITT, A. Minimal overhead virtualization of a large scale supercomputer. In *Proceedings of the 2011 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2011)* (March 2011).

[5] LANGE, J., PEDRETTI, K., HUDSON, T., DINDA, P., CUI, Z., XIA, L., BRIDGES, P., GOCKE, A., JACONETTE, S., LEVENHAGEN, M., AND BRIGHTWELL, R. Palacios and kitten: New high performance operating systems for scalable virtualized and native supercomputing. In *Proceedings of the 24th IEEE International Parallel and Distributed Processing Symposium* (April 2010).